TURBO news

Revista para Computadores ATARI Nº 3. Septiembre 1989

$ 550

- CURSOS: BASIC - LOGO - ASSEMBLER
- DESARROLLANDO HARDWARE
  - Construye tus Paddles

DESPACHO A REGIONES I - II - XI y XII $ 40



ADQUIERELOS EN LOS SIGUIENTES PUNTOS DE VENTAS

• **ANTOFAGASTA:** COOPERCARAB / KW VIDEO / LA ESPAÑOLA • **VIÑA DEL MAR:** FALABELLA VIÑA / INSIS / MPR COMPUTACION • **VALPARAISO:** COMPUTRONIC • **SANTIAGO:** AUDIO BICICLETA INTERNAC / CASA ROYAL / CENTRO ATARI / COMERCIAL ESTADO / COMPUMANQUE / COMPUCENTER / FALABELLA AHUMADA / FALABELLA P. ARAUCO / IMACO / INFOGROUP / PC STORE / PETERSEN / ROLEC / SUPERMERCADOS UNIMARC / TASCO / VIDEO CLUB INTERNACIONAL • **RANCAGUA:** CASA ZUÑIGA • **CURICO:** MULTIHOGAR • **TALCA:** LIBRERIA "EL AHORRO" / MULTICENTRO / VIDEO CLUB CASSAL • **CHILLAN:** CASA EDISON • **CONCEPCION:** COOPERCARAB / DISMAR / DISMAR 2 / EQUS / PHANTER / RAPSODIA / SESCO • **LOS ANGELES:** DISTRIBUIDORA MERINO • **ANGOL:** SCORPIO • **VICTORIA:** CASA SIGMUND • **TEMUCO:** COMERCIAL MANQUEHUE / ESTABLECIMIENTOS GEJMAN / FALABELLA • **PUCON:** ELTIT • **VILLARRICA:** JOYERIA KETTERER • **VALDIVIA:** ELECTROMUSICA • **LA UNION:** IMPORTADORA COSMOS • **OSORNO:** CASA REAL / FOTO EXPRESS • **PUERTO VARAS:** ELECTRO HORN • **PUERTO MONTT:** COMERCIAL MANQUEHUE / DIMARSA • **COYHAIQUE:** FACI HOGAR • **PUNTA ARENAS:** BALFER LTDA.

CONTENIDO



Circulación mensual. Nacional e Internacional
Destinada a los usuarios de computadores ATARI (R) como material didáctico de PROGRAMACION TURBO news (R) es una publicación de EDITORA TURBO LTDA. Domicilio Av. Holanda 2456 - Teléfono: 2238063 SANTIAGO CHILE.

GERENTE: Arturo Valdivieso Martínez. **EDITOR:** Editora Turbo Limitada. **PRODUCCION:** Pedro P. Caraball Alvarez - Marcelo A. Waldbaum Olszevicki - Mauro Pieressa Schachtel, Programadores y Diseñadores de Computación **COLABORACION:** Silvia Edelstein. **PERIODISTA:** Rodrigo Manríquez. **ARTE/DISEÑO:** Odalí Guerrero López. **FOTOGRAFIAS:** Isabel Bellalta.

DEPARTAMENTO DE VENTAS Y PUBLICIDAD: Arturo Valdivieso M. Hernán Vittini G. Agradecemos la colaboración de: Coelsa S.A. Centro Atari (Augusto Leguía Sur 75). ATARI, es marca registrada de Atari Corporation. TURBO, es marca registrada de EDITORA TURBO LIMITADA (Reg. Marc. N°342428 9-05-89). Impresa en los talleres de Impresos Nova Ltda. quien actúa como Impresora.

EDITORIAL

Hola amigos. Saludamos a quienes, en nuestro números anteriores, ya nos conocieron y a los apasionados por la computación que hoy ven por primera vez nuestra revista.

Para los que se integran a esta comunidad atariana, no está demás reiterar que nuestra meta es difundir el conocimiento de la informática en forma amena y entretenida.

Hemos recibido cartas que nos alientan a seguir avanzando entre los chips y además nos confirman las inquietudes que ustedes, nuestros jóvenes lectores, tienen.

En estos días, de regreso ya de unas merecidas vacaciones de invierno, queremos plantear una reflexión sobre la voluntad que es necesario tener para avanzar en las tareas que ahora desarrollas. En algún tiempo más, serás tú quien deba elegir la actividad en que quieras desempeñarte.

Nosotros queremos apoyarte con lo que nos corresponde como medio de comunicación. Servir de foro para tus opiniones e informarte sobre las herramientas, que en todas la áreas, te da la computación.

Por eso, en este número presentamos un artículo sobre la función Antic, que es fundamental en el sistema operativo Atari. También se estudia el sistema de conectores. Además continuamos con las secciones presentadas en ediciones anteriores.

Para la próxima te anticipamos el tema de las comunicaciones y las funciones del Modem. No dejes de hacernos saber tus sugerencias y opiniones, para que esta sea siempre "tu Revista".



UN PROGRAMA INGENIOSO

LA VIDA DE PEDRO PABLO CARABALL A.

El es un fanático de la pantalla de televisión... pero siempre que esté conectada a un ATARI.

Pasa noches completas sin dormir para saciar la pasión de su vida: programar computadores.



"Soy un ave nocturna -dice- y me entretengo tanto en esto que ni siquiera me doy cuenta cuando sale el sol". Recién allí lo inunda el sueño. Duerme unas horas y después, nuevamente, se sienta frente a la pantalla.

Así es Pedro Caraball Alvarez, uno de los programadores más destacados de nuestro país.

Vive holgadamente en el sector de Colón, al oriente de nuestra capital. Tiene 27 años, viste ropa deportiva, es bastante informal y su exitosa carrera es un ejemplo de lo que se puede lograr cuando, con imaginación y perseverancia, uno se dedica a desentrañar los "secretos" de un computador.

El Atari te da plena libertad

Todo comenzó allá por el año 81, cuando en las manos de Pedro cayó una calculadora programable. Era de un amigo, pero Pedro fue quien descubrió como usarla y, lo que es más importante, como sacarle provecho. Metía "torpedos" en la memoria del aparato.

Más tarde descubrió el Atari que tenía su hermano y comenzó a dirigir hacia allí sus intereses.

Poco a poco fue seducido por los encantos que esconde la máquina. "Recuerdo que empecé con cosas muy simples, que ahora encuentro hasta bobas. Por ejemplo yo le pedía al computador un dibujito con un círculo en la pantalla y me salía el dibujito. Me fascinaba que la pantalla me entregara justo lo que le estaba pidiendo. Si yo quería llenar el circuito con color, la máquina lo hacía".

Actualmente Pedro se desempeña como programador y escritor en esta revista y es un "free lance" en el desarrollo de software.

Ha creado innumerables programas, todos bien cotizados en el mercado y los especialistas. Se destacan, entre sus producciones, los programas TURBO TENIS, ODISEA, SOC, etc...

Lo que no le falta a Pedro son las ideas. Hoy está trabajando en un sistema de alarma -para ATARI 800- que controle toda la casa. "Es sencillo -expresa-, es cosa de conectar los interruptores y las entradas de la casa al computador, de manera que si entra alguien, el aparato se dé cuenta enseguida".

También tiene pensado trabajar en algunos desarrollos de Hardware, "hacer que el computador se conecte a otras máquinas. Conectarlo, por ejemplo, a un torno, para que el ATARI haga la pieza, en vez de que haya un operario laborando allí".

Una vez, hasta construyó un sismógrafo acoplado al computador.

Para él, en todo caso, esto es sólo un juego que lo entretiene de sobremanera.

Cuando le dió la primera "fiebre" por el ATARI, su madre no le daba mucha importancia. Lo veía casi todo el día sentado frente a la pantalla y pensaba que sólo era la distracción de su hijo. Pero era algo más que eso. Era además su futura fuente de ingresos.

Con el Atari 800 se podrían controlar los semáforos de Santiago

"El mundo del computador es muy misterioso -dice- y dentro de él hay un universo impresionante de conocimientos. Mientras uno más se mete, más se da cuenta de lo poco que sabe acerca de él".

Se demoró 5 años en "desenmascarar" por completo la potencialidad del ATARI 800. Y dice, sin ocultar su orgullo: "el ATARI chico (800) los conozco completo. No hay cosa que no le haya descubierto".

Con facilidad este modelo puede manejar algo de lo más complicado. La Supervisión de la señalización del Metro le queda chica. "Sobrado, fácilmente lo haría, aunque no está construído para ello, sino que para cumplir funciones domésticas".

Para que nos hagamos una idea del grado de complejidad del ATARI 800: sin problema podría mantener funcionando más de mil semáforos de Santiago, controlando el tiempo de cada uno y la sincronización entre ellos.

Esto es lo que hace el computador chico. "Imagínate ahora lo que ofrece el ATARI-ST, es indescriptible, es otro mundo el 'ST', el grande... Uuuuy -exclama Caraball- me quedan años por delante para poder llegar a decir que lo conozco como al chico".

Es mi trabajo, mi juego, mi desafío

Es un romántico de la informática. Guarda como buen recuerdo el computador que lo inició en este vasto campo. "Fue un ATARI 400 -rememora nostálgico-, todavía lo tengo y lo uso a veces, para hacer cosas chicas. Es muy viejo, pero sigue dando buenos resultados".

En todo caso la vida de Pedro, no está exenta de preocupaciones. Comienza a desarrollar un programa y puede que no se lo compren. También puede suceder que sus desvelos sean en vano, pues a veces no se da en el clavo y no "aparece" el programa preciso que se intenta crear.

Pero en la mayoría de las ocasiones la pantalla le sonríe. "Así es mi trabajo -dice-, es un desafío y a la vez un juego mágico".



Aunque resulte paradojal, no aconseja que sigan su camino. Recomienda a los aficionados que aprendan a manejar programas, a sacarles el máximo provecho. Pero agrega: "a no ser que





UN PROGRAMA INGENIOSO

LA VIDA DE PEDRO PABLO CARABALL A.

quieran ser programadores. Si quieren que la máquina haga lo que ustedes quieran, entonces aprendan a programar".

La molestia mayor la presenta el tiempo que demora fabricar un programa. "Te puedes demorar meses en llegar a un programa



para realizar una tarea específica y después te encuentras con que el programa ya está hecho. Por eso es mejor, para el principiante, aprender a "exprimir" los que ya existen".

"En Chile, los programadores están perdidos -se queja-, a no ser que vayan a parar a un banco o algo así".

Y tiene sus razones para decirlo. "En Estados Unidos, un programa, por viejo y malo que sea, vale unos 6.000 pesos. Aquí, te lo venden hasta por 200 pesos. Si incluso el diario te ofrecen un cassette lleno por sólo mil pesos. Es cosa de sacar la cuenta, cada cassette trae más o menos 20 programas".

La única posibilidad para nosotros es hacer cosas para empresas, programas que sean caros y que no sea negocio importarlos.

Este es el rubro en el cual se quiere desarrollar. El inmenso campo de la industria.

Pero "hay poca confianza por parte del industrial chileno hacia los programas creados en Chile". Los hombres que manejan los capitales y las inversiones en nuestro país, se muestran más receptivos frente a la tecnología japonesa o norteamericana.

Lo que no saben, o mejor dicho, lo que no quieren reconocer, es que en este rincón del mundo hay hombres que conocen y que dominan el tema.

Pedro Caraball cree que "sería muy fácil reemplazar operarios por máquinas, todo a bajo costo y con la seguridad de que la máquina no va a fallar".

"La máquina puede hacer la misma labor, más rápido, con mayor seguridad y si se daña, se cambia. y reponer máquinas es sencillo".

Así están las cosas, y si se abre más este mercado, el horizonte que avizorarán los "fanáticos de la pantalla" será muy amplio.

"En todo caso -dice Pedro- sólo lograrán su objetivo los programadores que hagan cosas a otro nivel".

Como las que hace Pedro Caraball.

Continuando con el curso de Basic, daremos en este número las instrucciones necesarias para poder almacenar y recuperar los programas digitados. También presentamos otro programa explicado, para que puedas practicar todo lo enseñado hasta el momento.







Ya hemos dado en los números anteriores las instrucciones necesarias para poder hacer programas sencillos. Habrás notado que cada vez que apagas el computador todo lo realizado hasta el momento se pierde y si deseas volver a ejecutar cualquier programa debes volver a ingresarlo. Para evitar hacer esto, existen los medios de almacenamiento como la cassettera o la disquetera. La segunda es mucho más rápida en el proceso de grabación y lectura de los programas pero es un periférico mucho más caro. Otra diferencia importante es que con el cassette, para cargar un programa, debes buscarlo manualmente dentro del cassette para saber dónde grabar o dónde leer el programa deseado. Esa búsqueda se puede realizar anotando el número de cuentavueltas que poseen todos los grabadores y llevar así un control de dónde está cada programa. En el caso de la disquetera, cada programa puede identificarse por medio de un nombre y a través de éste,el computador automáticamente puede encontrar los programas y/o el lugar donde haya espacio en el disquette para guardarlo.

En el caso de tener un grabador, las instrucciones necesarias son las siguientes:

CSAVE

Esta instrucción se ejecuta en el modo inmediato, es decir, sin número de línea. Una vez ejecutada escucharás dos beeps. Deberás oprimir las teclas RECORD y PLAY de su grabador y oprimir la tecla RETURN de tu computador para comenzar la grabación. La instrucción para leer la información es:

CLOAD

También se ejecuta en el modo inmediato. Al ingresarla oirás un solo beep, oprime la tecla PLAY de tu grabador y la tecla RETURN de tu computador para comenzar la lectura.

Si tienes una disquetera, la misma viene acompañada de un disquette llamado DOS (Sistema operativo de disco). Para poder usarlo,debes encender el computador con la disquetera prendida y el disco DOS puesto. Al comenzar,la disquetera le "avisará" al computador que puede trabajar con ella. La cassettera no necesita avisarle. Siempre, aún teniendo la disquetera, se puede utilizar el grabador. Las instrucciones correspondientes son:

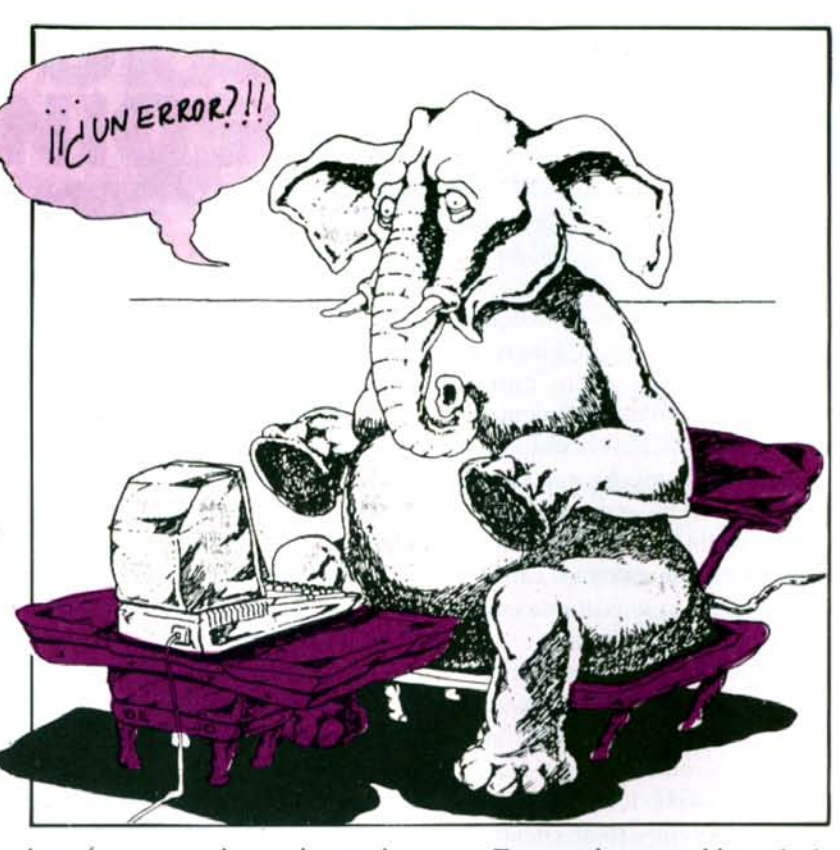
LOAD "D:nombre"

Al igual que en los casos anteriores, se ejecuta en el modo inmediato y se utiliza para cargar un programa desde el disco.

ASIC

TERCERA PARTE





BASIC

Nombre es cualquier palabra válida de como máximo ocho letras, más una extensión de tres letras optativa, separada con un punto, para darle mas claridad al nombre. Por ejemplo podría usarse la extensión .JUE para identificar a un juego o .UTI para un utilitario o .MAU para identificar al autor del programa, en este caso Mauro. Ejemplos:

```
LOAD "D:COMANDO.JUE"
LOAD "D:BANCOS.MAU"
LOAD "D:CAJA.UTI"
```

SAVE "D:nombre"

Es la instrucción contraria a LOAD y se utiliza para almacenar la información en disco.

A continuación vamos a entregar un programa que hace que el computador funcione como una calculadora. Previamente te daremos algunas instrucciones más, que te serán útiles, no sólo para dicho programa sinó para cualquiera que desees desarrollar.

INSTRUCCION TRAP número

Se utiliza para indicarle al computador a qué número de línea debe ir en caso de producirse cualquier tipo de error. En un mismo programa puede aparecer varias veces con números distintos según sea necesario. En nuestro programa lo utilizaremos para asegurarnos que se ingresen los datos en forma correcta. Esto es importante,porque como vimos en los número anteriores, si, cuando el computador nos solicita ingresar un número ingresamos una letra,daría un error y el programa se detendría. Con el TRAP logramos que en caso de producirse el error,el programa lo corrija solo. Ejemplo:

```
10 TRAP 20
20 INPUT NUM
30 TRAP 40
40 INPUT EDAD
```

En caso de error al introducir el valor de NUM, hará que se repita la operación de ingresarle el valor. Lo mismo ocurre al ingresar la edad.

Habrás notado lo desagradable que queda, en la ejecución de un programa, cuando no se borra la pantalla antes de que el programa comience a funcionar. Esto se puede solucionar incorporando a tu programa la instrucción:

BA

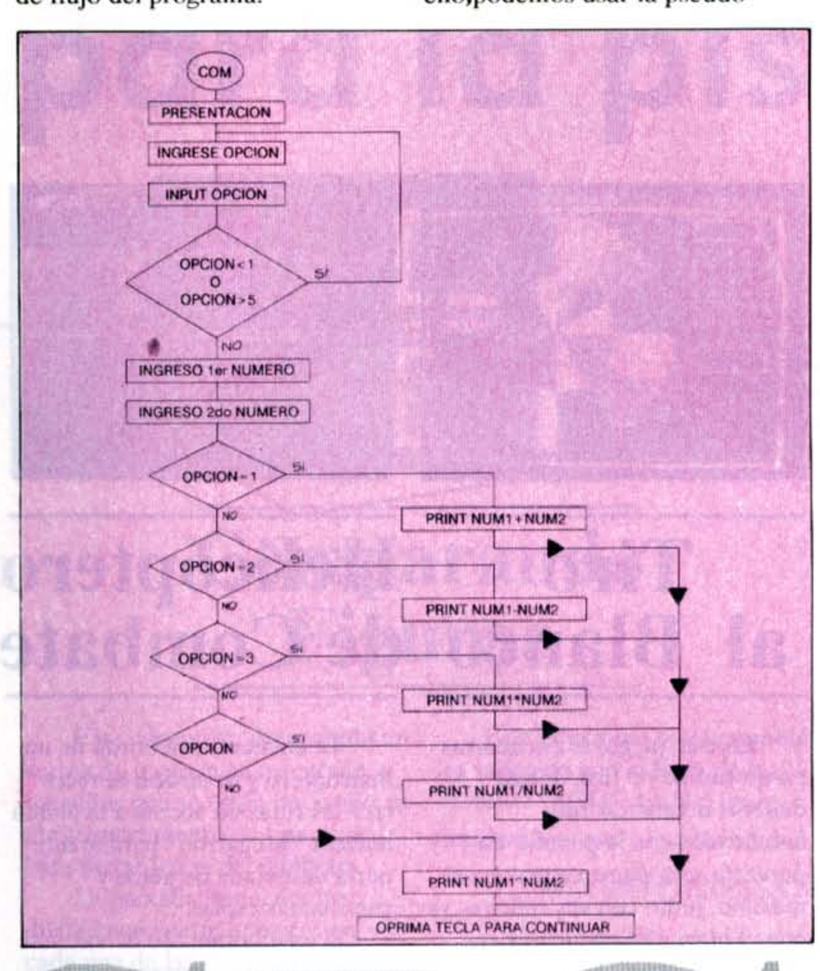
PRINT "¶"

Ese caracter lo conseguirás oprimiendo primero la tecla ESC y luego simultáneamente las teclas CONTROL y CLEAR. Al ejecutar esta instrucción, todo lo que hay en la pantalla se borrará.

Comencemos con el diagrama de flujo del programa.

Recuerda que para parar el programa debes oprimir la tecla BREAK.

Habrás notado que en el diagrama existen instrucciones que no son del Basic. Los diagramas no tienen que ser una copia fiel de lo que va a ser el programa. Estos sirven como guía. Por ello,podemos usar la pseudo-instrucción "PRESENTACION" que luego en la programación va a representar todo el grupo de instrucciones correspondientes. Inclusive este diagrama podría reducirse mucho más uniendo instrucciones graficadas en forma lineal (que no tienen ninguna pregunta en el medio). Por ejemplo,las pseudo-instrucciones INGRESE NUMERO1 e INGRESE NUMERO2 podría reemplazarse por una única que dijera INGRESE NUMEROS.



Analicemos el diagrama: La primer parte está dedicada al despliegue del menú principal. En computación se habla de "menú", cuando aparecen en pantalla diversas opciones de las cuales hay que seleccionar alguna. Allí validamos que no se haya seleccionado una opción menor a uno o mayor a cinco ya que cinco son las opciones posibles. A continuación ingresamos los dos números con los cuales vamos a operar. Finalmente, según cuál haya sido la opción elegida, imprimimos el resultado correspondiente. Para recomenzar, hacemos un INPUT cualquiera, cuyo contenido no nos interesa, pero nos sirve para detener el programa todo el tiempo que queramos. Al ingresar cualquier número y oprimir RETURN el programa comenzará desde el principio. Existen otras técnicas más prolijas para hacer esto y las iremos viendo en los sucesivos números. El listado del mismo lo encontrarás en la parte de programas al final de la revista.

En el próximo número continuaremos develando los secretos del Basic.

MAURO PIERESSA S.



TERCERA PARTE

PROGR

Como habíamos prometido en los números anteriores, seguimos entregándoles las novedades que la empresa M.P.M. S.A. destina a la pistola Atari, a través de su línea de productos TURBO SOFTWARE. En esta ocasión presentamos otros dos juegos, "Tiro al Blanco" y "Helicóptero de Combate", y un educativo, "Comparando Figuras".





Tiro al Blanco

En este juego, te encuentras en un campo de tiro, donde deberás demostrar tus habilidades con la pistola. Tu puntería sera puesta a prueba al máximo, junto con sus reflejos, ya que el blanco se encuentra en continuo movimiento.

Cada sector del blanco posee un color determinado asociado a un puntaje. Dichos colores van rotando continuamente a distintas velocidades, dependiendo del nivel que hayas escogido. Además obtendrás un puntaje extra en caso de acertarle al área del centro. Cuentas con treinta balas para lograr el mayor puntaje posible y vencer así a tus amigos.

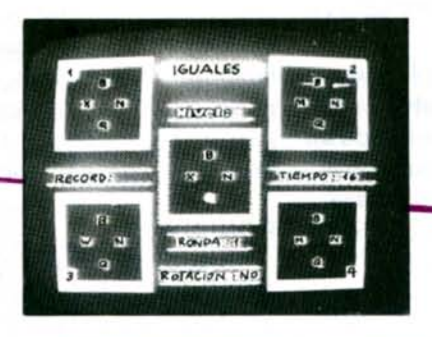
Helicóptero de Combate

Te encuentras a bordo de un helicóptero y tu misión es recorrer las rutas de acceso a la planta nuclear "Megatrón" para mantenerla despejada de autos y camionetas espías.

Si en el momento de vaciarse el cargador, has logrado por lo menos un total de 25 puntos, pasarás de nivel. Esto implica que deberás defender una zona perimetral más cercana a la base, donde la urgencia es mayor, debiéndote esforzar al máximo para continuar ascendiendo. En caso de no poder cumplir alguna etapa deberás recomenzar desde el principio.

RAMAS

para la pistola



Comparando Figuras

Este educativo, presentado en forma de juego, está orientado a los niños que se están iniciando en el reconocimiento de figuras o bien de las letras del alfabeto.

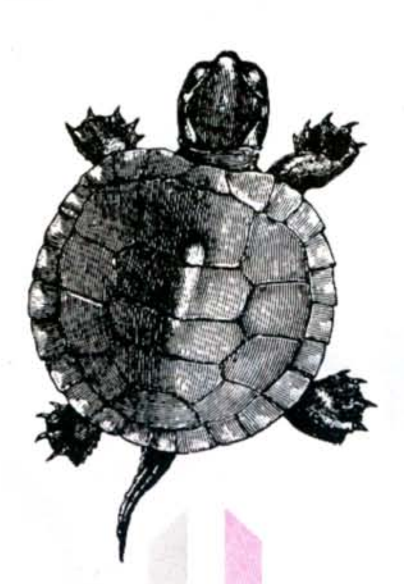
La pantalla se encuentra dividida en cuatro sectores en cada una de las esquinas de su televisor más un quinto sector resaltado en el centro. Cada sector puede contener un número determinado de figuras o letras, dependiendo del nivel escogido. Sólo uno de los cuatro sectores es idéntico al ubicado en el centro y será al que deberás disparar obteniendo un puntaje en función al tiempo en que se demoró la respuesta.

Este educativo puede también ser utilizado con el lápiz de luz: TURBO LIGHT-PEN o bien, simplemente con el teclado.

MAURO PIERESSA S.



aprendiendo a dibu LOG



Continuamos en este número con el curso de Logo dirigido a los más pequeños. Recordamos que la idea de este artículo es que los niños comiencen a dar órdenes al computador y ver su resultado. Por ello se está utilizando el lenguaje Logo únicamente en lo que respecta a la graficación.

Ya hemos visto en el número anterior algunas instrucciones para hacer nuestros primeros dibujos. En este capítulo agregaremos algunas más, sobre todo en el aspecto del color.

Comenzaremos explicando que cada una de las cuatro "TORTUGAS" tiene asociado un lápiz que le permite dibujar a su paso. Ese lápiz puede escogerse entre tres posibles,a los que hay que asignarle alguno de los 128 colores existentes. Si modificamos el color que le asignamos a algún lápiz,todo lo que éste haya dibujado cambiará de color. Para decirle a alguna tortuga qué lápiz va a usar se utiliza la instrucción PONNL. Cuando comenzamos a trabajar, las cuatro tortugas tienen asignado el lápiz 0 y cada uno de los tres lápices tiene un color definido.

INTRUCCION PONNL nro

Su significado es "Pon número de lápiz". Nro es un número que puede valer 0, 1 o 2 y representa cada uno de los tres lápices posibles. Ejemplo:

PONNL 1

Le asigna a la o las tortugas "despiertas" por la última instrucción DILE el lápiz número 1. Junto con esta instrucción se utiliza la que permita darle un color a dicho lápiz. La misma es:

INSTRUCCION PONLC nrolápiz nrocolor

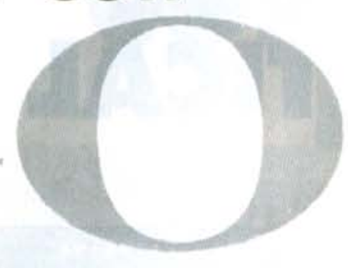
Su significado es "Ponle al lápiz color". Nrolápiz es el número del lápiz que queremos modificar. Como vimos, éste puede ser 1, 2 o 3. Nrocolor es un número que va del 0 al 127 y representa un color. Por ejemplo:

PONLC 0 20

Luego de ejecutarse esta instrucción, todo lo que dibujamos con el lápiz 0 cambiará al color naranja. A continuación damos un tabla con los colores y sus valores correspondientes. Notarás que a cada uno de los colores le corresponden ocho valores. La diferencia entre ellos está dada por la luminosidad, siendo los más bajos los mas oscuros y los más altos,los claros.

RANGO DE VALORES	COLOR
0 AL 7	GRIS
8 AL 15	AMARILLO
16 AL 23	NARANJA
24 AL 31	ROJO ANARANJADO
32 AL 39	ROSADO
40 AL 47	PURPURA
48 AL 55	PURPURA AZULADO
56 AL 63	AZUL OSCURO (CIAN)
64 AL 71	AZUL CIELO
72 AL 79	AZUL CLARO
80 AL 87	TURQUESA
88 AL 95	VERDE AZULADO
96 AL 103	VERDE
103 AL 111	VERDE AMARILLO
112 AL 119	NARANJA VERDOSO
120 AL 127	NARANJA CLARO

ar con



INSTRUCCION PONFD nro

Su significado es "Ponfondo" y nro,en un número que va del 0 al 127, que puede ser tomado de la tabla anterior. Se utiliza para asignar un color de fondo a la pantalla. Ejemplo:

PONFD 100

Luego de ejecutar esta instrucción,la pantalla se tornará verde.

A continuación,veremos la aplicación de estas instrucciones en la graficación de una casa. Tipea estas instrucciones oprimiendo la tecla Return a continuación de cada una de ellas y verás como avanza la construcción de la misma.

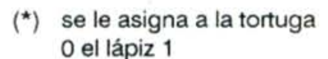
Al comenzar la tortuga 0 utiliza el lápiz 0 y el color amarillo

```
LM
AV 60
IZ 90
AV 100
IZ 90
AV 60
IZ 90
AV 100
IZ 90
AV 60
IZ 45
PONNL 1 (*)
PONLC 1 42 (**)
AV 30
IZ 45
AV 56
IZ 45
AV 30
RE 30
IZ 90
AV 30
PONNL 0 (***)
DE 45
AV 58
```

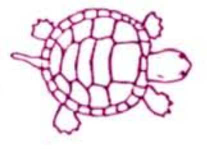
(*) se le asigna a la tortuga 0 el lápiz 1

(**) se le asigna al lápiz 1 el color 42 (rojo)

(***)se le reasigna a la tortuga 0 el lápiz 0



Una vez finalizado el gráfico, se puede hacer desaparecer la tortuga para que el mismo se vea con mayor claridad. Para ello, existen las siguientes instrucciones:



INSTRUCCION ST

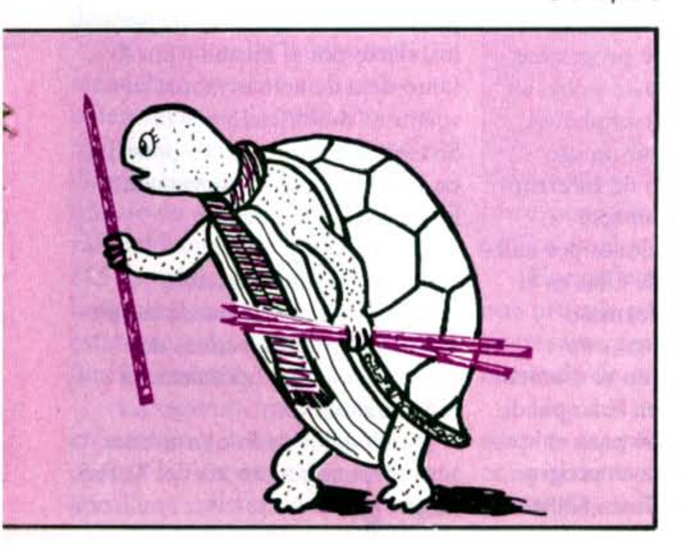
Su significado es "sin tortuga". Una vez ejecutada,verás como la tortuga desaparece. No obstante puedes seguir dibujando solamente con el lápiz. Podrás comprobarlo ejecutando a continuación de esta instrucción un, AV 30, por ejemplo, y verás que de todas maneras aparece una línea.

INSTRUCCION TG

Es la inversa de ST y su significado es "tortuga". Luego de ejecutada esta instrucción, verás como la tortuga reaparece.

Te desafiamos a completar la casa incorporándole, con otro color y utilizando el lápiz 2, una ventana y una puerta. Recuerda que con la instrucción SL, puedes desplazarte sin dibujar y con LA volverás a hacerlo.

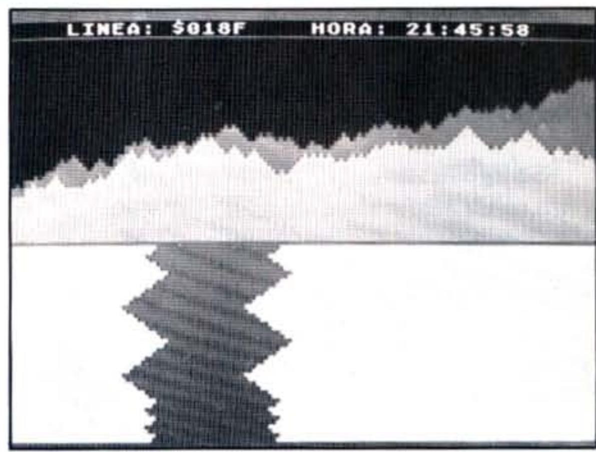
MAURO PIERESSA S.



INTERRUPCION DE BLANQUEO VERTICAL

ATARI POR DENTRO

por: PEDRO P. CARABALL A.



En esta oportunidad veremos otra función de ANTIC, la generación de la Interrupción de Blanqueo Vertical (Vertical Blank Interrupt), una de las bases de tiempo más importante de el Sistema Operativo del computador ATARI.

En esta interrupción están basados los traspasos de registros en Ram a los registros de hardware y viceversa.

Algunas de las funciones principales que cumple el Sistema Operativo en esta Interrupción son las siguientes:

- El Sistema Operativo le indica a Antic dónde se encuentra la lista de despliegue.
- Los registros de colores son traspasados a sus respectivos registros en el hardware.
- Se ejecuta la lectura y traspaso a Ram de los valores de joysticks, paddles, etc.
- Se genera el reloj del computador (ver Mapa de Memoria).
- Transferencia de información desde la grabadora al computador en el cálculo de baud rate (bits por segundo), en conjunto con el conteo de líneas horizontales.

En resumen, las funciones de esta Interrupción son muchas y muy importantes. En el buen uso de ella, se pueden basar programas completos. Por ejemplo, el TURBO TENIS es un programa que se ejecuta totalmente durante esta Interrupción. Aunque su elaboración fue algo más compleja que la de un programa de ejecución directa, los resultados obtenidos habrían sido imposibles sin el uso de esta Interrupción. Por eso sugerimos que cualquier rutina que deba compartir el tiempo de proceso con otro programa, sea escrita basándose en esta Interrupción (excepto algún caso especial, que no veremos en esta ocasión).

En la sección de programas hay un listado en Basic y otro en Assembler (ambos completos), que permiten analizar un uso practico de este tipo de Interrupción. En este programa,se ejecutan dos operaciones por cada Interrupción. Una de ellas es el cálculo de línea en formato hexadecimal en la cual esta trabajando el Basic en el momento de la Interrupción. Esto puede resultar muy practico para el seguimiento de algunos programas. El número de línea $8000 corresponde al modo directo (operación sin número de línea).

La otra operación, es mantener un reloj de tiempo real de 24 horas permanentemente en la pantalla, con despliegue de horas, minutos y segundos.

Estas dos rutinas necesitan la memoria disponible entre $600 y $6FF (1536 - 1791). El funcionamiento de ellas es permanente y seguro. Si la tecla Reset es presionada,la rutina no vuelve a instalarse por sí misma y por lo tanto deja de actuar,ya que su vector es modificado por el Sistema Operativo. Para ponerla en funcionamiento nuevamente basta con la instrucción X=USR(1536).

Con esta información, esperamos que Ud. pueda incluir en sus programas, rutinas de Interrupción que optimicen su funcionamiento.

Cualquier duda o problema será respondida por vía del Turbo Mail.

CONSTRUYENDO PADDLES

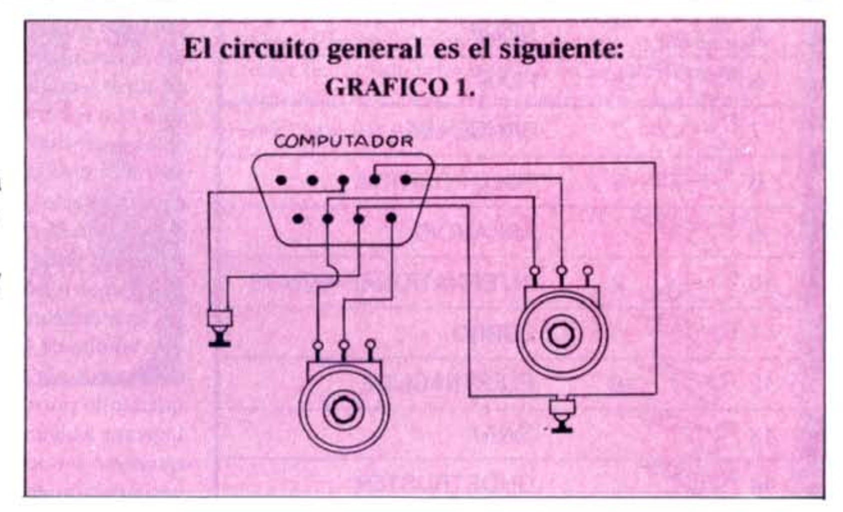
DESARROLLANDO HARDWARE

por: PEDRO P. CARABALL A.

En esta ocasión construiremos un par de paddles con el objeto de utilizar los juegos creados para este comando o para crear nuestros propios programas comandados por esta vía.

El circuito es en realidad muy sencillo y fácil de construir. Los materiales son baratos y los encontraremos en cualquier casa de artículos electrónicos.

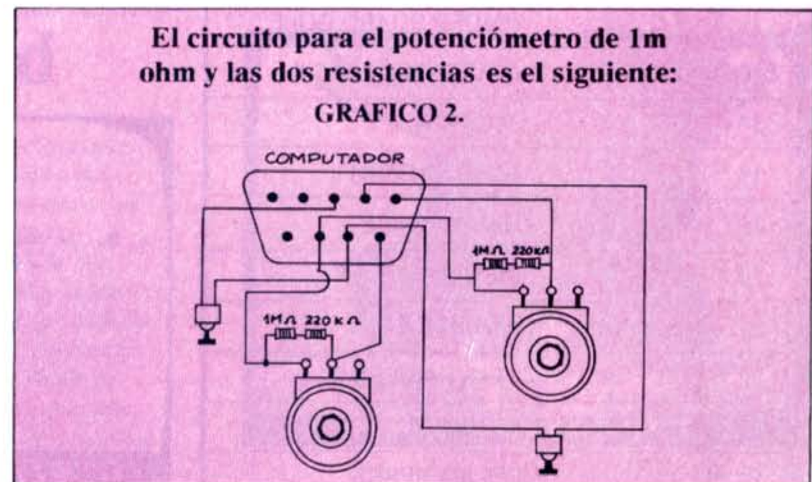
El circuito general es el siguiente:
GRAFICO 1.



Lista de materiales (2 paddles)

2 potenciómetros lineales de 550 k ohms.
2 pulsadores normalmente abiertos.
1 enchufe de joystick.
2 cables de 4 conectores.

El circuito para el potenciómetro de 1m ohm y las dos resistencias es el siguiente:
GRAFICO 2.



Dado que el valor del potenciómetro no es estándar, tenemos dos soluciones posibles:

La primera (y la más recomendable) consiste en usar un potenciómetro de 500k ohms de tipo lineal. Esto producirá una disminución en el rango de trabajo de nuestro paddle, variando desde un valor 1 hasta 215 aproximadamente, lo cual en la mayoría de los casos no establece ninguna diferencia en su funcionamiento.

La segunda opción es cambiar el potenciómetro por uno de 1m ohm (1 millón de ohms) de tipo lineal,una resistencia de 1m ohm, y una de 220k ohms. La conección correspondiente a esta variante se explica en el gráfico 2.

Es una buena idea terminar este artículo proponiendo un uso interesante para este nuevo comando de su computador.

Una de las formas más simples de utilizar este comando es en el control del cursor en un modo gráfico. Por ejemplo GR.24:PLOT PADDLE(0), PADDLE(1)/1.2, controlando los ejes X e Y de la posición del punto en el cual se dibujará.

Los botones o gatillos se leen mediante la instrucción PTRIG(N) donde N es el paddle a leer (de 0 a 3).

Espero que este circuito de construcción de paddle les proporcione utilidad, entretención y aprendizaje.

RANKING DEL MES

POS. DEL MES	POS. MES ANTERIOR	TITULO
1	4	SCREEMING WINGS 1942
2	3	MONTEZUMA REVENGE
3	9	GREAT AMERICAN RACE
4	10	RIVER RAID
5	1	NINJA
6	15	FLAK
7	7	BRUCE LEE
8	8	POLE POSITION
9	-	ARKANOID
10	2	INTERNATIONAL KARATE
11	-	ZORRO
12	16	ELEKTRAGLIDE
13	-	SWAT
14	-	GHOSTBUSTER
15	12	LEADER BOARD GOLF
16	-	WHO DARES WINS II
17	28	PACMAN
18	19	BOUDER DASH II
19	6	MIRAX FORCE
20	-	OIL'S WELL
21	11	HARDBALL
22	24	STAR RAIDERS II
23	5	FUTBOL II
24	-	POLAR PIERRE
25	-	RAID OVER MOSCOW

ASCENSO CONSTANTE DESCENSO

Este es el ranking correspondiente al mes de Julio, obtenido en base a las estadísticas de ventas de cassettes Turbo Software en todo Chile. Recuerda que tus preferencias también serán tenidas en cuenta, para lo cual podrás escribir a Holanda 2456, Providencia, con los juegos de tu elección.

Great American Race

En este juego participas en un exigente rally a través de todo el territorio de los EE.UU. En su recorrido deberás transitar los más variados paisajes, zonas nevadas, con neblina, zonas de baches y por supuesto el tránsito normal de las super-carreteras. No debes descuidar el consumo de combustible ni el esfuerzo a que está siendo sometido el motor, debiendo para ello hacer un buen uso de la caja de cambios. Para realizar un cambio deberás mover la palanca hacia arriba si deseas pasar a uno más alto, y para abajo en caso contrario. El botón del joystick funciona como acelerador. Si deseas frenar, debes hacerlo con un rebaje, es decir pasando a un cambio más bajo. En la parte superior de la pantalla, van apareciendo mensajes, que indican si estás cerca de la ciudad destino, si hay una bomba de bencina cercana, si se fundió el motor, o si se quedo sin combustible. Cuando notes que va quedando poco combustible, debes desacelerar para ingresar a alguna bomba. En caso de fundir el motor o quedarte sin bencina deberás "empujar" el auto hasta una estación de servicio. Esto lo lograras oprimiendo reiteradamente el botón del joystick.

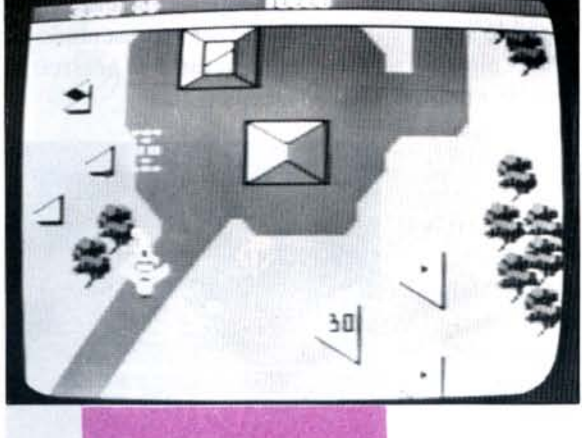


SONIDO	6.6
GRAFICACION	6.6
ADICCION	7
PRESENTACION	6.5
PROMEDIO	6.675

Flak



SONIDO	6.5
GRAFICACION	6.5
ADICCION	6.8
PRESENTACION	6.7
PROMEDIO	6.625



Es el año 2087 y el planeta Tierra se encuentra en guerra. Al mando de tu nave-cohete, encabezarás la invasión del territorio ocupado por el enemigo. Tu misión consiste en destruir todas sus defensas realizando para ello un vuelo a baja altura, abriéndole el camino a las fuerzas terrestres. Dichas defensas se encuentran fuertemente protegidas por corazas de una nueva aleación de acero, por lo que deberás aprovechar el momento exacto en que las mismas se abren para tratar de introducirles bombas en su interior y así destruirlas. Cuentas con una mira telescópica para poder apuntar a tu objetivo.

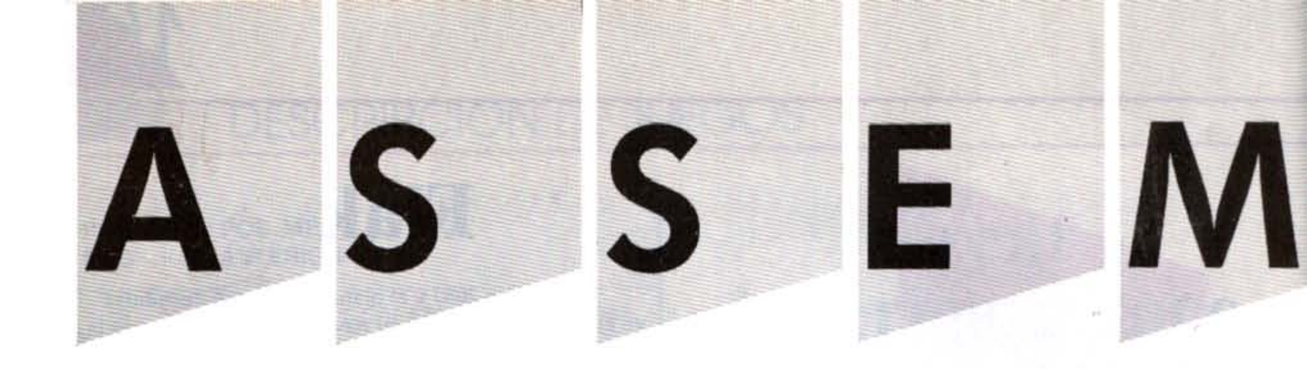
River Raid



Uno de los clásicos y siempre vigentes juegos para Atari. En este juego, controlas, por medio del joystick, un avión de combate que debe sobrevolar un angosto río en vuelo razante, mientras procuras destruir la mayor cantidad de blancos posibles. No debes descuidar la defensa, ya que el enemigo intentará derribarte a cualquier costo. Los objetivos que deberás enfrentar y destruir son: buques de guerra, aviones de combate, helicópteros, tanques y puentes. También encontrarás en el recorrido depósitos de combustible que podrás utilizar para recargar tus depósitos pasando sobre ellos. Dos jugadores pueden alternarse en esta excitante aventura, comenzando por el nivel de dificultad de su elección.

MAURO PIERESSA S.

SONIDO	6.2
GRAFICACION	6.4
ADICCION	6.9
PRESENTACION	6.4
PROMEDIO	6.475



Uno de los conceptos más difíciles de comprender en el desarrollo de programas en Lenguaje Assembler, es la utilización de las operaciones matemáticas simples. En este artículo, incluiremos todos los conceptos necesarios para realizar sumas y restas y en las siguientes ediciones de tu Revista TURBO news, desarrollaremos las restantes operaciones matemáticas.

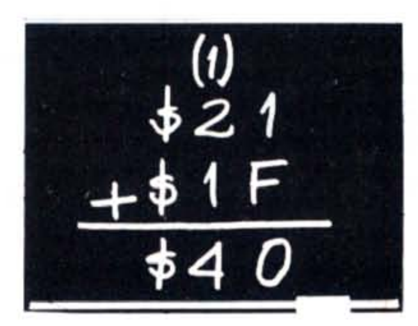
En aritmética decimal, sumar dos números de un dígito es fácil, siempre y cuando el resultado de la operación sea un número de un dígito. Sin embargo, si sólo contamos con un dígito para representar el resultado, la operación no resulta tan sencilla. Para comprender esta idea, desarrollemos el siguiente ejemplo. Supongamos que tenemos que ejecutar la operación 4+8. Si para representar el resultado disponemos de un solo dígito, la respuesta sería 2. Sin embargo, en matemática tradicional, esta operación genera un acarreo de 1.

(1)
04
+08
12

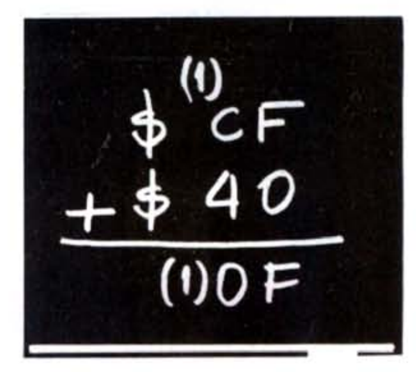
Es importante mencionar que el acarreo solamente puede ser igual a uno o a cero, pues ninguna combinación de valores de un solo dígito proporciona un acarreo mayor de 1.

En el caso de las operaciones binarias, este proceso de acarreo también se produce. Como el procesador de nuestro ATARI es de 8 bits, puede trabajar directamente con valores comprendidos entre 0 y 255. El computador necesita interpretar números mayores a 255, como nosotros necesitamos utilizar números mayores a un dígito. Nuestro sistema agrega un nuevo dígito, mientras el computador añade un nuevo byte.

Al igual que en las operaciones decimales, en las binarias, cuando su resultado no puede ser representado por un byte, el acarreo es igual a 1. Para dar un ejemplo de la suma binaria realizaremos las operaciones \$21 + \$1F y \$CF + \$40.



En esta operación, el resultado \$40 puede ser representado en un byte, por lo tanto el acarreo final es igual a 0.



En este caso, al sumar \$C con \$4 se produce el acarreo en 1. Este ejemplo de suma de dos bytes requiere para su representación de dos bytes, uno para contener el \$01 y el otro para el \$0F.

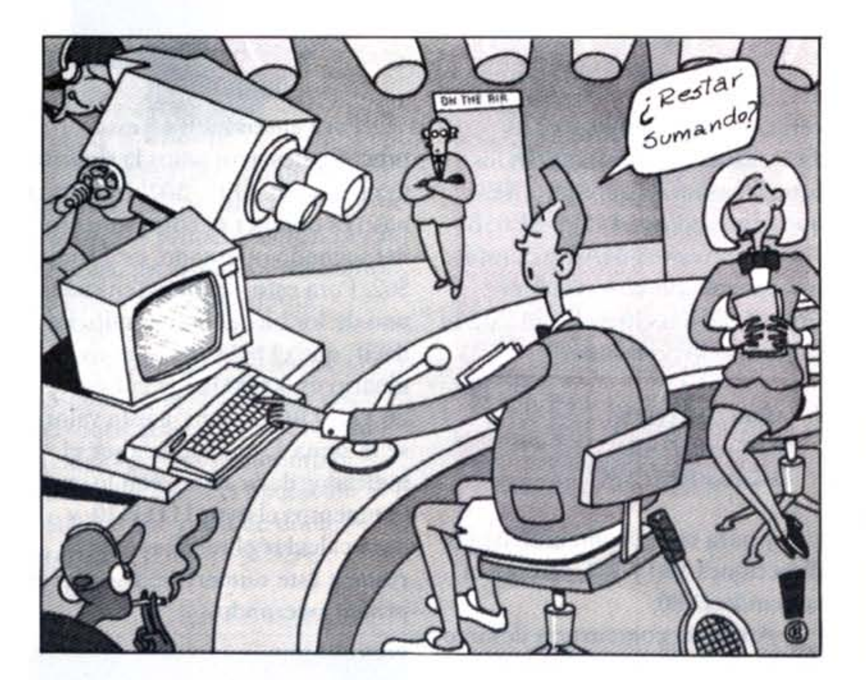
En el procesador 6502, existe un registro de uso especial llamado Registro de Estado, el cual contiene siete bits o flags para propósitos diferentes:

Bit -	7	6	5	4	3	2	1	0	-
Flag-	N	V		B	D	I	Z	C	-

N: indicador de resultado negativo
V: indicador de overflow
B: indicador de break
D: indicador decimal
I: indicador de interrupción
Z: indicador de resultado cero
C: indicador de acarreo

B L E R

TERCERA PARTE



El indicador de acarreo es utilizado en las operaciones de sumas y restas. Cuando se produce un acarreo en la operación, este bit se setea en 1. De lo contrario, se mantiene en el valor 0. Esta operación es automática, es decir, al realizar la suma o la resta, el procesador define este valor. Alguno de estos indicadores pueden setearse con instrucciones especiales. En el caso del indicador de acarreo, las instrucciones CLC y SEC se utilizan para definirlo.

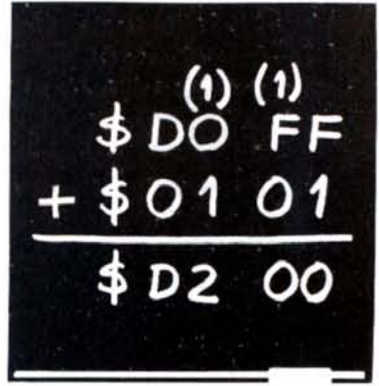
En el caso de la instrucción CLC, al ejecutarla, el procesador coloca un cero en el bit de acarreo y con la instrucción SEC se define al bit de acarreo en 1.

En el caso de sumar dos bytes cuyo resultado puede representarse en un byte deberemos realizar el siguiente listado:

```
100  .OPT OBJ
110  *=$4000
120 SUMA
130  CLC
140  LDA #$A1
150  ADC #$02
160  BRK
```

Con esta subrutina, el procesador primero ejecuta la instrucción CLC. Con ella, el indicador de acarreo se setea en 0. Luego, con la instrucción LDA #$A1, el acumulador se define con el valor $A1 y al ejecutar la instrucción ADC #$02 se le suma al acumulador el valor #$02, quedando el Registro A con el resultado de la suma de $A1 + $02. Es importante recordar que antes de efectuar cualquier suma, es necesario definir el bit de acarreo en cero, pues al ejecutar la instrucción ADC, el valor del acarreo es sumado al resultado.

Si queremos sumar en Assembler dos números cuyo resultado excede el valor 255, deberemos separar el resultado en dos bytes. Supongamos que deseamos realizar la operación $D0FF + $0101. Para desarrollarla manualmente efectuamos lo siguiente:



En esta operación, al efectuar primero la suma de los bytes de bajo orden, es decir, $FF + $01, el resultado obtenido es $00, pero con el acarreo igual a uno. Por lo tanto, el acarreo de esta suma debe sumarse al resultado de la suma de los bytes de alto orden. Entonces, al realizar $D0 + $01, le incorporamos el acarreo de $01 y obtenemos el resultado $D2. En Assembler, esta operación es automática. Como la operación ADC, define el valor del acarreo y además, suma a los operandos el



ASSEMBLER

acarreo anterior, lo único que tenemos que hacer es definir el indicador de acarreo en cero, sumar los bytes de bajo orden y luego los de alto orden, pero sin borrar el acarreo generado en la suma de los LO-BYTES. Si queremos realizar la operación $D01F + $103C en Assembler, deberemos ingresar el siguiente listado.

```
100  .OPT OBJ
110  *=$4000
120  CLC
130  LDA #$1F
140  ADC #$3C
150  STA $80
160  LDA #$D0
170  ADC #$10
180  STA $81
190  BRK
```

Para ejecutar este programa, es necesario ensamblarlo. Tal como lo explicamos en nuestra Lección Nro. 2, en la edición de Agosto, debemos ingresar el listado en el MAC/65, y luego ejecutar los comandos:

ASM y presionar RETURN
DDT y presionar RETURN
*=$4000 y presionar RETURN
y presionar la tecla START.

Una vez ejecutado el programa, el resultado de la suma quedará almacenado en las direcciones de memoria $80 y $81. Para verlos, podrás ingresar el comando * 80 y al presionar la tecla RETURN en tu monitor verás el contenido de ambas direcciones, las cuales serán los bytes hexadecimales $5B y $E0 que representan el valor $E05B.

Si en vez del MAC/65, estás trabajando con el Assembler Editor, deberás ingresar el listado y ejecutar los comandos:

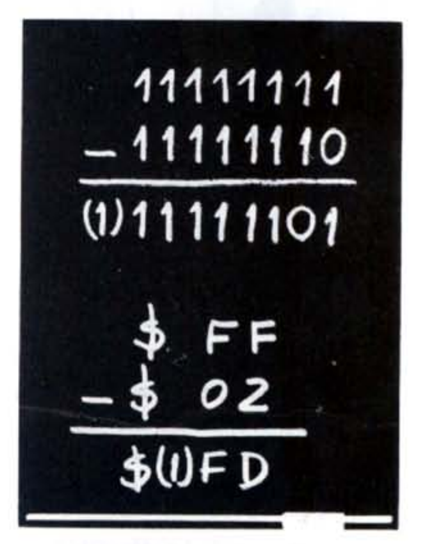
ASM y RETURN
BUG y RETURN
G4000 y RETURN

y para ver el contenido de las direcciones $80 y $81, ejecuta el comando D 80.

Antes de comenzar a desarrollar las Restas en Assembler, consideremos una revisión de la Instrucción ADC. El acumulador está cargado con un valor. Entonces el valor de una posición de memoria determinada se añade al mismo. El valor del indicador de acarreo 0 o 1 se suma a dicho valor y el resultado permanece en el acumulador. Si la suma se puede representar con un sólo byte, se pone a cero el indicador de acarreo. Si no es posible, el indicador de acarreo se pone en 1 para su uso por una posible suma de precisión mayor a la de un byte.

En el caso de las Restas, el procesador de nuestro ATARI aplica el sistema de suma de números complementados a dos. Esto es así, pues es más fácil diseñar un procesador que sólo sume y no realice operaciones de restas.

Para entender bien este proceso, desarrollemos la siguiente operación: $FF - $02. El primer paso es obtener el complemento del segundo operando, es decir, $02. Para esto, se invierten cada uno de los bits del operando. Es decir, que el byte $02, que en binario es 00000010 se convierte en 11111101. Luego, a este valor se le suma 1, que viene a ser el indicador de acarreo, con lo cual obtenemos el byte 11111110, y para calcular el resultado de la resta, a este número se le suma el primer operando.





Es importante recordar que antes de realizar una resta simple o antes del primer paso en restas de doble precisión, es necesario poner en 1 el indicador de acarreo.

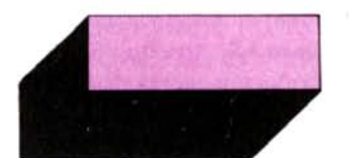
Si queremos restar en Assembler, la instrucción que debemos utilizar es SBC. Como ejemplo de una resta simple desarrollaremos la operación $FF - $02 y como resta de doble precisión $D01A - $032A.

RESTA DE PRECISION SIMPLE

```
100  OPT OBJ
110  *=$4000
120  SEC
130  LDA #$FF
140  SBC #$02
150  BRK
```

RESTA DE PRECISION DOBLE

```
100  .OPT OBJ
110  *=$4000
120  SEC
130  LDA #$1A
140  SBC #$2A
150  STA $80
160  LDA #$D0
170  SBC #$03
180  STA $81
190  BRK
```





Para ejecutar estos listados, deberemos procesar las mismas instrucciones que en los casos desarrollados para las sumas de simple y doble precisión.

En todos los listados de programas desarrollados en este artículo, hemos utilizado la instrucción BRK. Esta instrucción, que en el BASIC tiene su similar: STOP, es utilizada para detener la ejecución de un programa. Cuando el procesador se encuentra con esta instrucción, se detiene el programa y el control del procesador pasa a nuestras manos para realizar cualquier operación, en este caso, investigar el contenido de las posiciones de memoria $80 y $81.

En esta lección, hemos aprendido la utilización de las sumas y restas para nuestros futuros desarrollos en Assembler. En las siguientes ediciones de tu Revista TURBO news, te explicaremos como efectuar el resto de las operaciones matemáticas necesarias para realizar tus programas.

MARCELO WALDBAUM





LAS CONEC

por: PEDRO P. CARABALL A.

A pedido de nuestros lectores publicaremos sólo en este número, la información acerca de todos los enchufes de interconección que tiene el computador ATARI, exceptuando el bus de expansión y el conector de cartuchos por razones de espacio.

Comenzaremos por el conector para joystick, útil tambien para: paddle, track ball, mouse, light pen, light gun, touch table, modems, versa writer, laboratorios, etc.Como podemos ver,consta de 9 pines:

1 2 3 4 5
6 7 8 9

pin 1.- (joystick) adelante
pin 2.- (joystick) atrás
pin 3.- (joystick) izquierda
pin 4.- (joystick) derecha
pin 5.- potenciómetro B
pin 6.- botón
pin 7.- +5 volts
pin 8.- tierra
pin 9.- potenciómetro A

Utilizando este enchufe como entrada o salida de datos (pin 1 al 4) tenemos medio byte (nibble) de acceso paralelo. Nuestro computador posee 2 enchufes de libre uso, por lo cual disponemos de 8 bits (1 byte) de entrada/salida paralela con este sistema.

En la entrada de ambos potenciómetros A y B (pines 5 y 9) tenemos un conversor análogo digital, cuya función es trasladar el voltaje recibido a una representación numérica.

Cuando estos enchufes se encuentran programados como entrada, su valor es 1 por no tener conección alguna. Para obtener el valor cero en dicha entrada, es necesario conectar el pin correspondiente con la tierra (pin 8).

En el conversor análogo digital la situación es distinta. Al no tener ningun voltaje presente, la entrada nos entrega el valor 228. Al ir aumentando el voltaje, el registro correpondiente va disminuyendo, hasta 1 cuando el voltaje aplicado llega a los +5 volts (ver artículo: Desarrollando hardware, para la conección de un potenciómetro en esta entrada).

Para programar las entradas de joystick podemos utilizar este ejemplo:

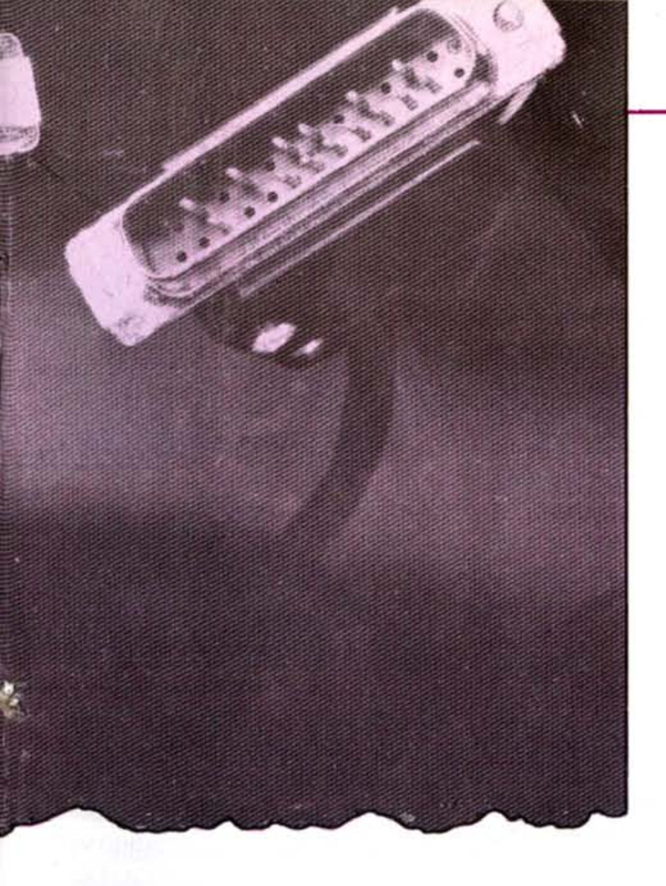
```
POKE 54018,48
POKE 54016,xxx
POKE 54018,60
```

Donde xxx está representado por esta tabla:

	$D300 (54016)							
	enchufe B				enchufe A			
valor	128	64	32	16	8	4	2	1
pin	4	3	2	1	4	3	2	1
entrada	0	0	0	0	0	0	0	0
salida	1	1	1	1	1	1	1	1

Para una correcta programación hay que sumar el Valor indicado arriba para los pines que usted desee programar como salida y luego seguir el ejemplo anterior de programación de entrada/salida con pokes.

Por ejemplo,si queremos programar los pines 1 y 2 del enchufe A y el pin 4 del enchufe B como Salida, los pines que no se



CIONES DEL ATARI

programan como salida quedan automáticamente como entrada, es decir debemos sumar 1+2+128 y luego ejecutar los tres pokes mencionados anteriormente y reemplazar xxx con el resultado recién obtenido, o sea: POKE 54016,131.

Para leer estas entradas,el formato es el siguiente:

X=PEEK(54016)

donde X refleja los valores de todos los bits puestos a uno, vistos en la tabla. Por ejemplo, si todos los pines están programados como entradas y X nos arroja el resultado 130 (128+2),tendremos que el pin 2 del enchufe A (valor 2) y el pin 4 del enchufe B (valor 128) están en uno y todos los demás en cero (conectados a tierra).

Una breve explicación del funcionamiento de la entrada del potenciómetro podría ser de gran ayuda para los más interesados.

El valor que encontramos en el registro de cada uno de los potenciómetros refleja la cantidad de líneas de barrido en el televisor que demora un capacitor en cargarse. Este se encuentra en serie con el potenciómetro, por lo tanto al bajar la resistencia girándolo,el capacitor requerirá menos tiempo para cargarse. Por ello, un valor menor será reflejado en el correspondiente registro.

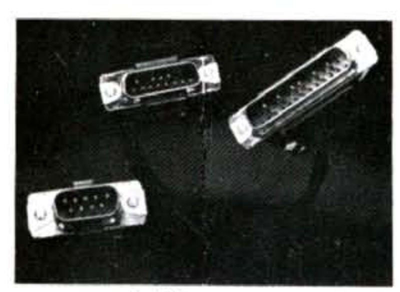
Para comenzar una lectura con los contadores en cero y los capacitores descargados es necesario escribir en el registro POTGO - $D20B (POKE 53771,255). Esto es efectuado por el Sistema Operativo en la segunta parte de la Interrupción de Blanqueo Vertical.

Para ejecutar una conversión rápida, la cual toma sólo dos líneas de barrido horizontal en lugar de las 228 utilizadas en el modo anterior, debemos realizar los siguientes pasos: el bit 1 (los bits se deben contar de 0 a 7) del registro SKCTL - $D20F, 53775 (control de puerta serial) controla el modo de trabajo de este conversor. Cuando este bit está puesto a cero, el modo de trabajo es el normal. Si está puesto a uno, el tiempo de carga máxima del capacitor es de 2 líneas de barrido. Hay que tener cuidado pues en este modo rápido los transistores ocupados de esa tarea, no descargan al capacitor. Por ello, debemos volver a poner este bit en cero para descargar dichos capacitores. Luego ponerlo nuevamente a uno para la siguiente lectura.

Debemos también tener en cuenta que,con el incremento de velocidad, existe una pérdida de precisión en la conversión.

Para completar la información acerca de este conector, hablaremos del light pen y/o light gun.

Cuando el pin 6 es conectado a tierra ANTIC toma el valor del contador vertical y lo deposita en el registro PENV - $D40D (54285), Para el calculo de la posición horizontal, en el mismo



LAS CONECCIONES DEL ATARI

momento es tomado el valor del contador de períodos de color y depositado en el registro PENH - $D40C (54284).

Hablaremos ahora del enchufe de entrada/salida (I/O) serial, utilizado en la conección de la grabadora, disketera, impresora, interfaces, etc.

Enchufe serial:

2 4 6 8 10 12
1 3 5 7 9 11 13

pin 1.- entrada de reloj
pin 2.- salida de reloj
pin 3.- entrada de datos (serial)
pin 4.- tierra
pin 5.- salida de datos (serial)
pin 6.- tierra
pin 7.- command
pin 8.- control de motor
pin 9.- proceed
pin 10.- +5/ready
pin 11.- entrada de audio
pin 12.- +12v (400-800)
pin 13.- interrupt

La puerta serial, denominación habitual que recibe este conector, consiste en:

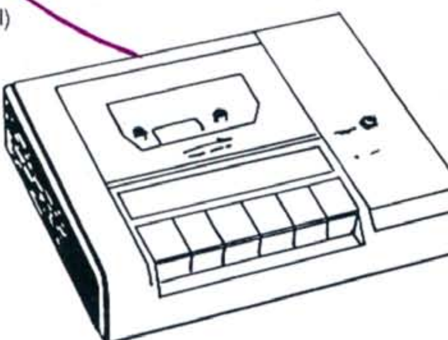
- Salida serial (transmisión)
- Entrada serial (recepción)
- Salida serial de reloj
- Línea serial bidireccional de reloj
- Líneas de control

La información es recibida y transmitida como 8 bits seriales precedidos por un cero lógico a modo de bit de comienzo y sucedidos por un uno lógico como bit de finalización.

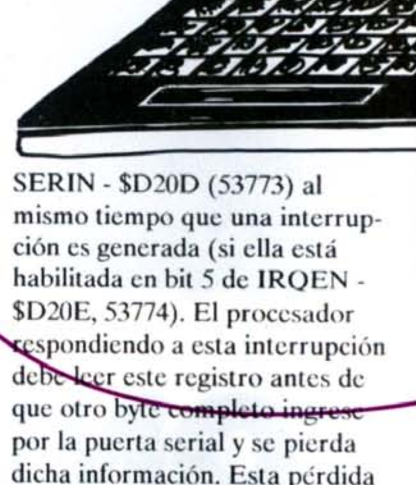
Los relojes de entrada y salida (pines 1 y 2) son iguales al baud rate (bits por segundo). La información transmitida cambia cuando el reloj de salida se pone en uno. La información recibida es sampleada (leída) cuando el reloj de entrada se pone en cero.

El procesador puede poner la línea de salida de información en cero. Para esto se debe poner en uno el bit 7 de SKCTL (control de puerta serial).

Reloj serial de salida (pin 2): la información de salida serial siempre cambia cuando este reloj se pone a uno. El reloj vuelve a cero en la mitad del tiempo del bit de salida. El baud rate del reloj y la información queda determinado por los canales de audio 2 y 4, o por el reloj de entrada, dependiendo del modo elegido (usando bits 4, 5 y 6 de SKCTL).



Entrada serial (pin 3): la información puede ser recibida por el procesador, leyendo el bit 4 de SKSTAT - $D20F (53775), (ignorando el registro de shift). Una vez recibidos los ocho bits de información más el bit de comienzo (start bit) y el bit de finalización (stop bit), la información es transferida automáticamente a



SERIN - $D20D (53773) al mismo tiempo que una interrupción es generada (si ella está habilitada en bit 5 de IRQEN - $D20E, 53774). El procesador respondiendo a esta interrupción debe leer este registro antes de que otro byte completo ingrese por la puerta serial y se pierda dicha información. Esta pérdida de información (en este caso: over-run) es detectada mediante la lectura del bit 6 de SKSTAT (cero indica error). El bit 7 de SKSTAT en cero indica error de bits perdidos o extra bits.

Reloj bidireccional (pin 2): puede ser usado como reloj de entrada o salida determinando el baud rate de la información, dependiendo de los bits 4, 5 y 6 de SKCTL.

Entrada serial asincrónica: este modo de trabajo de la puerta serial, permite leer información que no proporciona un reloj para determinar su baud rate. Esto se logra gracias a que el registro de shift puede ser comandado por el canal de audio 4. En este caso, los canales 3 y 4 deben ser usados en conjunto para lograr una mayor precisión. Esto se hace poniendo en uno el bit 3 de AUDCTL - $D208 (53768). El baud rate



puede variar hasta un 5% debido a que cada vez que se recibe un bit de comienzo de byte, los canales 3 y 4 son reseteados, permitiendo ésto que la información tenga una mínima diferencia de velocidad con la determinada por los canales 3 y 4.

Salida serial de datos (pin 5): al igual que la entrada serial debe hacerse mediante una interrupción. El bit de habilitación de dicha interrupción es el bit 4 de IRQEN y el vector se encuentra en $20C (524). El byte a transmitir debe ser dejado por la rutina de interrupción en el registro SEROUT - $D20D (53773), siendo éste traspasado al registro shift de salida. Los bits de comienzo y finalización son puestos automáticamente.

La información normalmente es transmitida como niveles lógicos (+4v=uno 0v=cero), pero puede también ser transmitida como dos tonos. En este caso, el canal uno es transmitido como el uno lógico y el canal dos, como el cero lógico. Para lograr ésto, debemos poner en uno el bit 3 de SKCTL.

Para programar el modo de trabajo de la puerta serial, se debe seguir la siguiente tabla:

000: transmisión : reloj externo
recepción : reloj externo

001: transmisión : reloj externo
recepción : asincrónica

010: transmisión : canal 4
recepción : canal 4
canal 4 en reloj de salida

011: no tiene utilidad

100: transmisión : canal 4
recepción : reloj externo

101: no tiene utilidad

110: transmisión : canal 2
recepción : canal 4
canal 4 en reloj de salida

111: transmisión : canal 2
recepción : asincrónica

En los dos últimos casos no puede utilizarse el modo de dos tonos ya que el canal 2 está siendo usado para determinar la velocidad de transmisión.

La linea de Command (pin 7) puede ser controlada mediante el bit 3 de PBCTL - $D303 (54019)

La línea de control de motor de la grabadora (pin 8), puede ser manejada mediante el bit 3 de PACTL - $D302 (54018). Para



prender el motor de su grabadora, puede tipear POKE 54018,52 y para apagarlo POKE 54018,60. Mediante este sistema, se puede poner un cassette de audio en su grabadora ATARI y escuchar la música por el parlante de su televisor o monitor.

Las líneas de proceed e interrupt (pines 9 y 13), no son usadas por el Sistema Operativo ni por ningún periférico ATARI. Estas líneas pueden ser utilizadas por ustedes siguiendo esta sencilla estructura:

Interrupt: produce una interrupción al procesador si el bit 0 de PACTL se encuentra en uno. Esta interrupción se produce en flanco de subida si el bit 1 de PACTL es uno y en flanco de bajada cuando el bit 1 de PACTL es cero. El vector de esta interrupción se encuentra en $204 (516). Para resetear esta interrupción, permitiendo así que la misma se repita, es necesario leer PORTA - $D300 (54016). Como el Sistema Operativo solo guarda el acumulador, antes de retornar el control desde este nivel, debemos poner la instrucción PLA y sólo entonces RTI.

Proceed: El funcionamiento de esta línea es igual al de



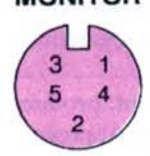


LAS CONECCIONES DEL ATARI

Interrupt, sólo que sus bits de habilitación se encuentran en PBCTL, su vector está en $202 (514). Para resetearlo es necesario leer PORTB - $D301 (54017).

Por último, describiremos el conector de MONITOR, que consta únicamente de salidas no programables, por lo que no es necesario detallar sus características.

MONITOR



pin 1.- luminancia compuesta
pin 2.- tierra
pin 3.- salida de audio
pin 4.- video compuesto
pin 5.- croma compuesta

Como conclusión final, creemos que esta información le servirá para conocer a fondo todos los detalles de los conectores de su computador.

La información entregada en estas páginas es el fruto de años de trabajo con el computador ATARI, el estudio de una gran cantidad de textos, manuales y circuitos, la cual es muy difícil de encontrar en alguna publicación en español.

Cumplimos así el objetivo de entregarle la mejor y más completa información acerca de su ATARI.

MAPA DE MEMORIA

por: PEDRO P. CARABALL A.

El mapa de memoria describe las posiciones de memoria y sus utilidades en la programación.

En esta nueva sección estudiaremos dicho mapa. Pondremos especial énfasis, en las posiciones de memoria más importantes para los programadores Basic.

Una posición de memoria es un lugar donde se guarda un número, éste puede ser un dato o parte de un programa.

Nuestro computador ATARI tiene 65536 posiciones de memoria (64k) las cuales se numeran desde 0 hasta 65535. En cada una de ellas,podemos dejar cualquier número que se encuentre en el rango de 0 a 255.

Algunas de estas posiciones, son utilizadas por el Sistema Operativo. Este es el programa que permanentemente controla las funciones de nuestro computador. Este lee los valores de las posiciones de memoria y los ocupa para determinadas funciones, las cuales analizaremos en detalle más adelante con el objeto de simplificar este mapa.

Sólo mostraremos las posiciones más importantes, que presenten una utilidad práctica de fácil implementación en sus programas, puesto que para mostrarlas todas necesitaríamos varias revistas completas.

Es recomendable para aquellos programadores que utilizan su computador en forma profesional,adquirir el libro MAPPING THE ATARI, que pueden encontrar en su Centro ATARI.

Pasamos entonces a detallar el mapa general de nuestro computador, para tener una idea global de la configuración del sistema:

MAPA GENERAL

POSICIONES DE MEMORIA:

desde	hasta	usado por
0	127	Sistema Operativo
128	255	Basic
		Paquete de coma flotante
256	511	Pila
512	1151	Sistema Operativo
		Buffer de Cassette
		Buffer de Impresora
1152	1405	Basic
1406	1535	Paquete de coma flotante
1536	1791	Disponible
1792	X	Sistema Operativo de Disco (DOS) (si está presente)
X	40959	X=PEEK(128)+PEEK(129)*256
		Disponible
		Gráficos
		Programas
40960	49151	Basic (rom)
49152	52223	Control de interrupciones (rom)
52224	53247	Caracteres internacionales
53248	53503	GTIA
53504	53759	Bus de expansión
53760	54015	POKEY
54016	54271	PIA
54272	54527	ANTIC
54528	54783	Linea control catridge
54784	55295	Rom no usado?
55296	57343	Paquete de coma flotante
		Bus de interfase paralela
57344	58367	Set de caracteres
58368	65535	Sistema Operativo

Para leer una Posición de Memoria debemos usar la instrucción PEEK. Por ejemplo,si deseamos saber qué valor tiene la posición 632, el comando adecuado sería PRINT PEEK(632). Si queremos modificar una posición de memoria y poner en ella un número que nosotros elegimos, debemos usar la instrucción POKE. Por ejemplo, si queremos poner el número 148 en la posición 712, el comando correcto sería POKE 712,148.

Veamos ahora algunas Posiciones de interés general y su uso adecuado:

16 **Tecla BREAK:** Para desconectar esta tecla debemos usar los siguientes comandos: POKE 16,112:POKE 53774,112

18,19,20 **Reloj del computador:** Para cálculo de tiempo debemos usar la siguiente fórmula: TIEMPO=INT ((PEEK(18) *65536+PEEK(19)*256+PEEK(20))/ 60). Para mostrar la hora en modo "hora,minuto,segundo" se requiere un programa específico.

63 **Fin de archivo de cassette:** Si contiene un cero, significa que aún quedan datos, cualquier otro número indica que se ha encontrado el fin de archivo.

65 **Sonido de entrada/ salida (I/O):** Pokee esta dirección con un cero y el traspaso de información a los periféricos será silenciosa. Con otro número regresará el sonido.

77 **Cambio de color automático:** Si su programa no utiliza el teclado por más de 7 minutos, pokee esta dirección con cero periódicamente para evitar la rotación de colores. Para comenzar inmediatamente la rotación pokee 77 con 128.

82 **Margen izquierdo**: El Sistema Operativo pone aquí un 2. Si desea cambiar este margen, ponga aquí el número correspondiente al margen deseado. Ej: POKE 82,20 (20=número de columnas a la izquierda del margen.

83 Igual que 82 pero refleja la posición del margen derecho, inicializada en 39. Estas posiciones sólo son válidas para gráfico 0 y ventana de texto. Dicha ventana consiste en 4 líneas de gráfico cero en la parte inferior



MAPA DE MEMORIA

de otro modo gráfico, por ejemplo: gráfico 8.

84 **Posición de fila actual del cursor:** Cero corresponde a la parte superior de la pantalla.

85,86 **Posición de columna actual del cursor:** Cero corresponde a la posición izquierda de la pantalla. Debe usarse el siguiente cálculo:
COL = PEEK(85) + PEEK(86)*256

88,89 **Inicio de la pantalla en Ram:** Estas posiciones determinan en que parte de la Ram se encuetra el primer byte (parte superior izquierda) de la pantalla. Para imprimir el caracter A con poke usar la siguiente instrucción:
POKE PEEK(88) + PEEK (89)*256,ASC("A")-32.

106 Cantidad de Ram en páginas (una pagina = 256 bytes). Para ubicar el fin de la memoria Ram use la siguiente instrucción: RAM = PEEK(106)*256. Para reservar Memoria debe usar la siguiente instrucción: POKE 106,PEEK(106)-número de paginas a reservár.

En el próximo número de nuestra revista,veremos algunos usos prácticos para estas posiciones de memoria y agregaremos algunas más a nuestro mapa.

Con estos elementos, Ud. ya puede familiarizarse con el mundo de PEEK y POKE, logrando con ello un mejor dominio de su computador ATARI.

TURBO MAIL

En esta nueva sección de tu Revista TURBO news, encontrarás la respuesta a todas tus inquietudes respecto al computador ATARI. Si deseas formularnos alguna pregunta, escribenos a Editora Turbo Limitada, Holanda 2456, Santiago, para que podamos publicar tu carta y nuestra correspondiente respuesta.

MANTENTE EN CONTACTO

Sres. "TURBO news":
Mi nombre es Cristian Salgado Herrera. Primero que nada deseo felicitarlos por su entretenida y útil revista (una de las pocas). Tengo un ATARI 800-XL y mi duda es como controlar por software, la salida de +5V de los joysticks. Ademas me gustaría contar con una lista de libros donde se encuentren explicados los pokes.

Cristian:
Como podrás ver, en esta edición de TURBO news incluímos un artículo en el cual se explican todos los conectores que tu ATARI posee. Asi que leyéndolo, podrás satisfacer tu inquietud respecto al conector del joystick.

Como material de lectura, te podemos recomendar el libro MAPPING THE ATARI, el cual contiene una completa explicación acerca de todas las direcciones de memoria. Además podrás notar que incorporamos en nuestra Revista, una sección especial que describe el mapa de memoria de tu computador ATARI, la cual será publicada en todas las ediciones de TURBO news.

Sres. "TURBO news":

Me voy a presentar, soy Andrés Plaza, un usuario fiel de ATARI y mi carta tiene variados fines.

El primero es felicitarlos por una revista EXCELENTE, con la calidad de una revista norteamericana, buenos reportajes, entretenida, con muchas novedades. Realmente la única revista super completa para ATARI. Al principio, por la portada pensé que sería de puros juegos, pero me sorprendí cuando la leí. Todo es muy entretenido, especialmente el desarrollo de hardware.

Para el Turbo Mail, me gustaría saber si Uds. tienen programas de comunicaciones, como para hacer un BBS con el Modem XM-801. También podrían fomentar con un artículo, las comunicaciones vía modem, que están poco difundidas en Chile.

Para el ránking del mes, voto por dos programas: S.W.A.T. y Ghostbusters.

También me gustaría saber si en sus oficinas venden personalmente programas en disco, más bien utilitarios.

Mando también un disco con programas hechos por mí, para que los publiquen, si les gustan.

Por último, reitero mis felicitaciones por una revista EXCELENTE.

Andres:

Gracias por escribirnos. Para tu tranquilidad, en la proxima edición de TURBO news, encontrarás una sección especial, dedicada a promover las comunicaciones vía Modem. Tenemos pensado describir las características principales que un buen Modem debe poseer para su mejor rendimiento, así como tambíen analizar el software existente en el mercado.

En nuestras oficinas no comercializamos productos. Con respecto a tus programas, te informamos que en estos instantes nos estamos dedicando a analizarlos, junto a los demás discos que hemos recibido de otros lectores de TURBO.

Tambíen agradecemos tu colaboración en la obtención del ránking de este número y no olvides que tu software ingresó en el concurso de 10 suscripciones por 6 meses a tu revista TURBO news.

Estimados "TURBO news":

En primer lugar, quiero felicitarlos por su reciente y excelente revista, la cual creo que va a servir a muchos usuarios de computadores ATARI. Quisiera que publicaran algún artículo sobre el Modem telefónico y su software, como por ejemplo el XM-301 para 8 bits.

También quisiera que publicaran un Curso de TURBO BASIC, el cual es mucho más fácil que el Basic standard que se incluye en el ATARI. Se despide afectuosamente: Dennis Kangme Y.

Dennis:

En la respuesta a la carta de Andrés Plaza, le prometimos que en la próxima edición de TURBO news, incorporaremos una cobertura especial del tema de las comunicaciones vía Modem. Así que en el próximo número podrás obtener la información que nos solicitas.

Con respecto al TURBO BASIC, como podrás entender, partimos con un curso de Basic Standard, pues todos los usuarios de ATARI pueden acceder a él. En el caso del TURBO BASIC, es necesario contar con un periférico de almacenamiento como la unidad de discos o la cassettera. De todas formas, tenemos ya planeado, una vez que el curso standard evolucione, comenzar con la explicación de las ventajas que incorpora el TURBO BASIC a la estructura standard del lenguaje BASIC.

Curso Basic

```
1 REM ****************************
2 REM *   REVISTA TURBO NEWS (R)   *
3 REM *        MAURO PIERESSA      *
4 REM *        CURSO DE BASIC      *
5 REM *                            *
6 REM * OBJETIVO: Aplicacion de    *
7 REM * instrucciones en la crea-  *
8 REM * cion de una calculadora.   *
9 REM ****************************
10 ? "K          MENU PRINCIPAL"
20 ? "          ---------------"
30 ? :? :? :? "1) SUMA"
40 ? :? "2) RESTA"
50 ? :? "3) MULTIPLICACION"
60 ? :? "4) DIVISION"
70 ? :? "5) EXPONENCIACION":? :?
80 TRAP 80:? "INGRESE SU OPCION ";
90 INPUT OPCION
100 IF OPCION<1 OR OPCION>5 THEN 80
110 ? "KINGRESE PRIMER NUMERO";
120 INPUT NUM1
130 ? "INGRESE SEGUNDO NUMERO";
140 INPUT NUM2:? :?
150 IF OPCION=1 THEN ? "EL RESULTADO E
S ";NUM1+NUM2:GOTO 300
160 IF OPCION=2 THEN ? "EL RESULTADO E
S ";NUM1-NUM2:GOTO 300
170 IF OPCION=3 THEN ? "EL RESULTADO E
S ";NUM1*NUM2:GOTO 300
180 IF OPCION=4 THEN ? "EL RESULTADO E
S ";NUM1/NUM2:GOTO 300
190 IF OPCION=5 THEN ? "EL RESULTADO E
S ";NUM1^NUM2:GOTO 300
300 TRAP 300:? :? :? "OPRIMA UN NUMERO
 Y LA TECLA RETURN":? "          PARA CO
NTINUAR ";
310 INPUT AUX:GOTO 10
```

Atari por dentro

```
10 REM ****************************
20 REM *   REVISTA TURBO NEWS (R)   *
30 REM *    PEDRO P. CARABALL A.    *
40 REM *      ATARI POR DENTRO      *
50 REM *                            *
60 REM * OBJETIVO: demostracion de *
70 REM * uso de interrupcion de     *
80 REM * blanqueo vertical.         *
90 REM ****************************
99 REM  LECTURA DE DATOS
100 RESTORE 1000:CHEQUEO=0
110 TRAP 170:POKE 195,0
120 POKE 559,0:REM APAGAR LA PANTALLA
(30% mas de velocidad)
130 FOR DIRECCION=1536 TO 1789
140 READ DATO:CHEQUEO=CHEQUEO+DATO
150 POKE DIRECCION,DATO
160 NEXT DIRECCION
170 POKE 559,34:REM RECONECTAR LA PANT
ALLA
180 IF PEEK(195)=0 THEN 200:REM ERROR?
190 ? "ERROR ";PEEK(195):END :REM SI H
AY ERROR
200 IF CHEQUEO=22660 THEN 220:REM DATO
S CORRECTOS
210 ? "ERROR DE DATA":END :REM DATOS I
NCORRECTOS
219 REM : INGRESO DE HORA
220 DIM HORA$(8)
230 TRAP 230
240 ? "INGRESE HORA (hh,mm,ss) > ";
250 INPUT #16;HORA$
260 IF LEN(HORA$)<8 THEN 240
270 HORA=VAL(HORA$)
280 IF HORA>23 OR HORA<0 THEN 240
290 MINUTO=VAL(HORA$(4))
300 IF MINUTO>59 OR MINUTO<0 THEN 240
310 SEGUNDO=VAL(HORA$(7))
320 IF SEGUNDO>59 OR SEGUNDO<0 THEN 24
0
329 REM  TRASPASO DE HORA A PROGRAMA
```

```
330 FOR X=0 TO 6 STEP 3
340 POKE 1628+X,ASC(HORA$(X+1))-32
350 POKE 1629+X,ASC(HORA$(X+2))-32
360 NEXT X
370 X=USR(1536):REM CONECTRAR INTERRUP
CION
380 END
400 REM DATOS PROGRAMA
1000 DATA 104,24,173,48,2,133,208,105,
3,141
1010 DATA 63,6,173,49,2,133,209,105,0,
141
1020 DATA 64,6,160,255,200,177,208,201
,65,208
1030 DATA 249,200,169,57,145,208,141,4
8,2,200
1040 DATA 169,6,145,208,141,49,2,234,1
69,7
1050 DATA 169,122,162,6,76,92,228,112,
112,66
1060 DATA 65,6,1,255,255,0,0,0,0,0
1070 DATA 44,41,46,37,33,26,0,4,16,16
1080 DATA 16,16,0,0,0,0,40,47,50,33
1090 DATA 26,0,16,16,26,16,16,26,16,16
1100 DATA 0,0,0,0,0,16,17,18,19,20
1110 DATA 21,22,23,24,25,33,34,35,36,3
7
1120 DATA 38,0,173,49,2,201,6,240,13,1
69
1130 DATA 96,141,47,6,32,1,6,169,234,1
41
1140 DATA 47,6,238,121,6,173,121,6,201
,60
1150 DATA 208,67,169,0,141,121,6,162,7
,160
1160 DATA 2,254,92,6,189,92,6,201,26,2
08
1170 DATA 26,169,16,157,92,6,202,254,9
2,6
1180 DATA 189,92,6,201,22,208,10,169,1
6,157
1190 DATA 92,6,202,202,136,16,220,173,
92,6
1200 DATA 201,18,208,15,173,93,6,201,2
0,208
1210 DATA 8,169,16,141,92,6,141,93,6,1
60
1220 DATA 0,177,138,72,74,74,74,74,72,
200
1230 DATA 192,2,208,243,162,0,104,41,1
5,168
1240 DATA 185,105,6,157,78,6,232,224,4
,208
1250 DATA 241,76,98,228
```

LISTADO 2

```
0100 ******************************
0110 *                            *
0120 *    REVISTA TURBO NEWS (R)  *
0130 *                            *
0140 *      PEDRO P. CARABALL A.  *
0150 *                            *
0160 *                            *
0170 *       ATARI POR DENTRO     *
0180 *                            *
0190 * OBJETIVO: DEMOSTRACION USO *
0200 *           INTERRUPCION DE  *
0210 *           BLANQUEO VERTICAL *
0220 *                            *
0230 * nota: Este programa es solo *
0240 *       informativo, el mismo *
0250 *       esta incluido en el  *
0260 *       programa basic de este *
0270 *       mismo articulo.      *
0280 *                            *
0290 ******************************
0300 ;
0310     .OPT NO LIST
0320     .OPT OBJ
0330 ;
0340 RTS =    $60
0350 STMCUR = $8A
0360 DLIST = $D0
0370 NOP =    $EA
0380 SDLSTL = $0230
0390 SETVBV = $E45C
0400 XITVBV = $E462
0410 ;
0420     *=  $0600   ; 1536
```

```
0430 ;
0440      PLA           ; 0 argumentos
0450 REPONERDL
0460      CLC           ; para sumar
0470      LDA SDLSTL    ; puntero lista
0480      STA DLIST     ; de despliegue a
 $d0
0490      ADC #$03      ; saltar los 3
0500      STA OLDL      ; primeros bytes
0510      LDA SDLSTL+1 ; para saltar
0520      STA DLIST+1 ; desde la segund
a
0530      ADC #$00      ; lista de despli
egue
0540      STA OLDL+1    ; ---------------
----
0550      LDY #$FF      ; y = 255
0560 LOOP
0570      INY           ; comienzo en 0
0580      LDA (DLIST),Y ; buscar fin de
0590      CMP #$41      ; lista de despli
egue
0600      BNE LOOP      ; ---------------
----
0610      INY           ; encontrado
0620      LDA # <NEWDL ; apuntar
0630      STA (DLIST),Y ; vector a
0640      STA SDLSTL    ; comienzo de
0650      INY           ; la nueva
0660      LDA # >NEWDL ; lista de
0670      STA (DLIST),Y ; despliegue
0680      STA SDLSTL+1 ;---------------
-----
0690 RTSBYTE ;          en caso de volver
 a entrar
0700      NOP           ; poner rts, no s
etear VBI
0710 ;
0720 ;
0730 ; puesta en funcionamiento de int
errupcion
0740 ;
0750      LDA #$07      ; diferida
0760      LDY # <VBI    ; ubicacion de
0770      LDX # >VBI    ; rutina
0780      JMP SETVBV    ; activar y volve
r al basic
0790 ;
0800 ;
0810 ; nueva lista de despliegue
0820 ;
0830 NEWDL
0840      .BYTE $70     ; 8 lineas en bla
nco
0850      .BYTE $70     ; 8 lineas en bla
nco
0860      .BYTE $40+$02 ; lectura de da
tos en memoria + gr.0
0870      .WORD TEXTO ; ubicacion de lo
s datos
0880      .BYTE $01     ; saltar
0890 OLDL
0900      .WORD $FFFF ; antigua lista
0910 ;
0920 ;
0930 ; texto de la linea superior
0940 ;
0950 TEXTO
0960      .SBYTE "     LINEA: $"
0970 LINUM
0980      .SBYTE "0000     "
0990      .SBYTE "HORA: "
1000 TIMER
1010      .SBYTE "00:00:00      "
1020 ;
1030 ;
1040 ; conversion numero de linea a he
xadecimal
1050 ;
1060 HEX
1070      .SBYTE "0123456789ABCDEF"
1080 ;
1090 ;
1100 ; tiempo en 1/60 de segundo
1110 ;
1120 SESENTAVOS
1130      .BYTE 0
```

```
1140 ;
1150 ;
1160 ; runtina de interrupcion
1170 ; blanqueo vertical
1180 ;
1190 VBI
1200 ;
1210 ;
1220 ; verificacion de despliegue efec
tivo
1230 ; de nuestra linea superior en ca
so
1240 ; de cambio de grafico
1250 ;
1260      LDA SDLSTL+1 ; a=lista de dep
liegue
1270      CMP # >NEWDL ; a= nuestra lis
ta?
1280      BEQ OKNEWDL ; si, continue
1290      LDA #RTS     ; no, modifique b
yte
1300      STA RTSBYTE ; para no setear
VBI
1310      JSR REPONERDL ; reponga nuest
ra lista
1320      LDA #NOP     ; reponga byte No
 OPeracion
1330      STA RTSBYTE ; para nueva entr
ada via USR
1340 OKNEWDL
1350      INC SESENTAVOS ; sesentavos=s
esentavos+1
1360      LDA SESENTAVOS ; a=sesentavos
1370      CMP #60      ; a=60?
1380      BNE NO24HORAS ; no, no desple
gar tiempo
1390      LDA #$00     ; a=0
1400      STA SESENTAVOS ; sesentavos=a
1410      LDX #$07     ; 8 caracteres
1420      LDY #$02     ; 3 separaciones
1430 LOOPTIMER
1440      INC TIMER,X ; incremento en 1
1450      LDA TIMER,X ; seg,min,horas
1460      CMP #'9-31   ; pasamos de 9?
1470      BNE FINTIMER ; no, termino el
 despliegue
1480      LDA #'0-32   ; si, poner en 0
1490      STA TIMER,X ; donde correspon
da
1500      DEX          ; pasar a columna
 decenas
1510      INC TIMER,X ; sumar 1
1520      LDA TIMER,X ; decenas
1530      CMP #'6-32   ; pasaron de 6?
1540      BNE FINTIMER ; no, fin despli
egue
1550      LDA #'0-32   ; si, poner en 0
1560      STA TIMER,X ; donde correspon
da
1570      DEX          ; puntero en sepa
racion
1580      DEX          ; columna siguien
te
1590      DEY          ; separacion sigu
iente
1600      BPL LOOPTIMER ; tres veces
1610 FINTIMER
1620      LDA TIMER    ; leer decenas de
 horas
1630      CMP #'2-32   ; 20 horas?
1640      BNE NO24HORAS ; no, falta par
a 24 horas!
1650      LDA TIMER+1 ; si, leer unidad
es de horas
1660      CMP #'4-32   ; 4 horas?
1670      BNE NO24HORAS ; no, todavia f
alta!
1680      LDA #'0-32   ; 24 horas, volve
r a 00:00:00
1690      STA TIMER    ; decenas de hora
s
1700      STA TIMER+1 ; unidades de hor
as
1710 NO24HORAS ;      fin reloj
1720 ;
1730 ;
1740 ;despliegue numero de linea basic
1750 ;
```

```
1760      LDY #$00     ; primer byte
1770 LOOP1
1780      LDA (STMCUR),Y ; puntero line
a basic
1790      PHA          ; guardar
1800      LSR A        ; dividido 2
1810      LSR A        ; dividido 4
1820      LSR A        ; dividido 8
1830      LSR A        ; dividido 16
1840      PHA          ; guardar
1850      INY          ; proximo
1860      CPY #$02     ; solo 2 numeros
1870      BNE LOOP1    ; formato lo,hi
1880      LDX #$00     ; primer byte a d
esplegar
1890 LOOP2
1900      PLA          ; recuperar
1910      AND #$0F     ; nibble bajo
1920      TAY          ; puntero a tabla
 de conversion
1930      LDA HEX,Y    ; tomar valor hex
adecimal
1940      STA LINUM,X ; desplegarlo
1950      INX          ; proximo
1960      CPX #$04     ; estan los 4?
1970      BNE LOOP2    ; no, continuar
1980      JMP XITVBV   ; si, fin interru
pcion!
```

NUEVA

DISKETTERA ATARI XFF-551

En pocos segundos gane mucho tiempo

La diskettera Atari XFF-551, impone una nueva velocidad de trabajo al computador ATARI XL/XE Sólo unos pocos segundos bastan, para que la unidad de diskettes magnéticos de 5¼ pulgadas XFF cargue completamente un programa en la memoria de su computador Sin demoras, sin esperas, sin errores. Para un veloz acceso a la información que necesita, incorpórela rápidamente a su computador y pase adelante a toda prisa.







Sinónimo de garantía y servicio.

Ocho de cada diez computadores para el hogar, son ATARI